MAIN MENU MODULE

TECHNICAL REQUIREMENTS

Introduction

This document is the technical response to the Main Menu User Requirements document. It will describe how the development team will implement the changes and additions to SWSS Childrens to effect the requirements.

This document is also to be used as a tool by the development team when coding the solution or maintaining it in the future. Thus this document is likely to be updated during the lifecycle of the SWSS project. Versions of this document will be maintained in PVCS, and the reader should be aware that multiple printed versions may exist.

Module Description

This module gathers case information to be stored into the SWSS_INI.INI file before any other module is executed. It is the common point from which most modules are started most of the time. Several security checks, based upon who is supposedly logged onto the system, are made to determine if the user has access rights to a module, if any. These security checks are not all inclusive, however they offer a vast starting point. This module also activates/deactivates corrections mode for certain users.

The Main Menu module is also the preferred place for a user to end a SWSS "session".

The Main Menu module is also the only place that a user may activate or deactivate Corrections mode.

Requirements

Process Description

- This module must shut down after successfully "shelling out" to another module.
- The user may use an Alt+s <key> keystroke sequence. For example, the user may select the "Funding Determination" module by pressing the Alt and "s" keys and then the "f" key. This is a common Windows procedure for using the keyboard to select a menu item.
- This module will serve as a central point from which almost all SWSS modules can be accessed.
- There must be multiple mechanisms for a SWSS user to select a case before accessing a module.
  - If the log number is know, a "Log #" field,
  - Selecting the "Find A Client" (Soundex) button,
  - Selecting the "Case Listing" button,
  - Add a brand new case to the SWSS system with the "Add New" button (which accesses the Soundex module).
- If the user is actively working on a case and he/she returns to the Main Menu module, the log number of the selected case will pre-fill in the "Log #" field. Also, if the user selects a case from the case Listing or Soundex modules, the log number of the case which he/she has selected will again pre-fill the "Log #" field.
- This module is also the "clean" exit point for the user to terminate a SWSS "session".
- Correction mode can be activated/deactivate in this module.

Functional Requirements

1) The Main Menu module stores basic case information in the SWSS_INI.INI file. Each module then reads this information as the user loads it up. Some of this information includes:
   - Case Information
     - Log number
     - Case name (also, in most cases, the child's name)
     - Child ID
     - Current legal status
     - Case state
     - Case status
     - Case County
     - Load number of the primary worker for the case
     - Program code currently assigned to the case
     - Goal code
     - Case number (CIS number)
     - The user's basic update access rights to the case
     - A list of worker ID's that have full update access to the case
     - A list of worker ID's that have secondary access rights to the case
2) When the user leaves the SWSS application from this module, most of the data in the INI file is cleared.
3) The Main Menu module also attempts to clear up all memory resident objects when the user terminates a SWSS session.

Business Events

None

List of Program Units

This would be stuff like common VB code called, the number of VB .BAS modules and form modules in the current application. Also list the Stored Procedures called. Show a "Structure Diagram" of which VB procedure calls a stored procedure or another VB procedure. Also show which stored procedures call other stored procedures.

Visual Basic Code Components

Forms
frmAboutCFC (Aboutcfc.frm)
frmError (Error.frm)
frmInit (FRMINIT.FRM)
frmSplash (frmSplash.frm)

Code Modules
millerm7 (millerm7.bas)
swss (swss.bas)

SQL Stored Procedures and Packages
{SWSS.Main_Menu.get_case_info}
{SWSS.Main_Menu.get_responsibility}
{SWSS.Main_Menu.get_update_worker_ids}
{SWSS.Main_Menu.get_scndry_wrkr_Ids}
{SWSS.Main_Menu.get_sup_sec_wrkr}
{SWSS.Main_Menu.get_co_adpt_supvsr_id}

MODULE: FRMINIT.FRM

FIRST CALL DEPTH	SECOND CALL DEPTH	THIRD CALL DEPTH	FOURTH CALL DEPTH	FIFTH CALL DEPTH
CmdAdd_Click()	fLaunchSoundex()	swss. PrivateINI_Register( )		
		swss.tGetAppPath()		
		swss.lPrivateINI_PutString()		
		swss.LaunchEXE()		
cmdCaseListing_Click()	mnuScreensCaseListing_Click()	millerm7.TicklerShells()		
		Form_QueryUnload()		
		Form_Unload()	swss.CleanUpObjects()	
			swss.ClearINI_Sections()	
CmdFind_Click()	fLaunchSoundex()	swss. PrivateINI_Register( )		
		swss.tGetAppPath()		
		swss.lPrivateINI_PutString()		
		swss.LaunchEXE()		
	swss.PrivateINI_Register()			
	swss.tGetAppPath()			
	swss.bPrivateINI_SectionExists()			
	swss.tPrivateINI_GetString() (if swss.bPrivateINI_SectionExists returns True)			
fDeselectAllImgLblPairs(), fDeselectSurroundingControls()	fDeselectImgLblPair()			
Form_Load()	swss.PreviousInstance()			
	frmSplash.Show			
	swss.CenterControl()			
	swss.SetDeviceIndependentWindow()			
	swss.PrivateINI_Register()			
	swss.tGetAppPath()			
	swss.PrivateINI_DeleteSection()			
	fFadeForm()			
	swss.ExtractINI_Info()			

MODULE: FRMINIT.FRM				
FIRST CALL DEPTH	**SECOND CALL DEPTH**	**THIRD CALL DEPTH**	**FOURTH CALL DEPTH**	**FIFTH CALL DEPTH**
	swss.CheckForCorrections()			
	Form_QueryUnload()			
	Form_Unload()	swss.CleanUpObjects()		
		swss.ClearINI_Sections()		
	fLogID_Change()	millerm7.GetCaseInformation()		
	frmSplash.Form_QueryUnload()			
	frmSplash.Form_Unload()			
Form_MouseMove(), lblFrmInit_MouseMove(), lblInstructions_MouseMove(), picMenuIcons_MouseMove(), SSPanel1_MouseMove()	fDeselectAllImgLblPairs()	fDeselectImgLblPair()		
Form_Unload()	swss.CleanUpObjects()			
	swss.ClearINI_Sections()			
img5DayPacket_Click(), lbl5DayPacket_Click()	mnuScreens5DayPacket_Click()	millerm7.TicklerShells()		
		Form_QueryUnload()		
		Form_Unload()	swss.CleanUpObjects()	
			swss.ClearINI_Sections()	
img5DayPacket_MouseMove(), imgCaseClosing_MouseMove(), imgCaseRegistration_MouseMove(), imgCaseSummary_MouseMove(), imgChildInformation_MouseMove(), imgComments_MouseMove(), imgEducation_MouseMove(), ImgExit_MouseMove(), imgFundingDetermination_MouseMove(), ImgLegal_MouseMove(), imgMARE_Registration_MouseMove(),	fSelectImgLblPair()	fDeselectSurroundingControls()	fDeselectImgLblPair()	

MODULE: FRMINIT.FRM				
FIRST CALL DEPTH	**SECOND CALL DEPTH**	**THIRD CALL DEPTH**	**FOURTH CALL DEPTH**	**FIFTH CALL DEPTH**
imgMedicaid_MouseMove(), imgMedical_MouseMove(), imgMemberInformation_MouseMove(), imgPayment_MouseMove(), imgPlacement_MouseMove(), imgProvider_MouseMove(), imgReportGeneration_MouseMove(), imgUtilities_MouseMove(), lbl5DayPacket_MouseMove(), lblCaseClosing_MouseMove(), lblCaseRegistration_MouseMove(), lblCaseSummary_MouseMove(), lblChildInformation_MouseMove(), lblComments_MouseMove(), lblEducation_MouseMove(), lblExit_MouseMove(), lblFundingDetermination_MouseMove(), lblLegal_MouseMove(), lblMARE_Registration_MouseMove(), lblMedicaid_MouseMove(), lblMedical_MouseMove(), lblMemberInformation_MouseMove(), lblPayment_MouseMove(), lblPlacement_MouseMove(), lblProvider_MouseMove(), lblReportGeneration_MouseMove(), lblUtilities_MouseMove()				
imgCaseClosing_Click(), lblCaseClosing_Click()	mnuScreensCaseClosing_Click()	millerm7.TicklerShells()		
		Form_QueryUnload()		
		Form_Unload()	swss.CleanUpObjects()	
			swss.ClearINI_Sections()	

MODULE: FRMINIT.FRM				
FIRST CALL DEPTH	**SECOND CALL DEPTH**	**THIRD CALL DEPTH**	**FOURTH CALL DEPTH**	**FIFTH CALL DEPTH**
imgCaseRegistration_Click(), lblCaseRegistration_Click()	mnuScreensCaseRegistration_Click()	millerm7.TicklerShells()		
		Form_QueryUnload()		
		Form_Unload()	swss.CleanUpObjects()	
			swss.ClearINI_Sections()	
ImgCaseSummary_Click(), lblCaseSummary_Click()	mnuScreensCaseSummary_Click()	millerm7.TicklerShells()		
		Form_QueryUnload()		
		Form_Unload()	swss.CleanUpObjects()	
			swss.ClearINI_Sections()	
imgChildInformation_Click(), lblChildInformation_Click()	mnuScreensChildInfo_Click()	swss.PrivateINI_Register()		
		swss.tGetAppPath()		
		swss.lPrivateINI_PutString()		
		millerm7.TicklerShells()		
		Form_QueryUnload()		
		Form_Unload()	swss.CleanUpObjects()	
			swss.ClearINI_Sections()	
imgComments_Click(), lblComments_Click()	mnuScreensComments_Click()	millerm7.TicklerShells()		
		Form_QueryUnload()		
		Form_Unload()	swss.CleanUpObjects()	
			swss.ClearINI_Sections()	
imgEducation_Click(), lblEducation_Click()	mnuScreensEducation_Click()	millerm7.TicklerShells()		
		Form_QueryUnload()		
		Form_Unload()	swss.CleanUpObjects()	
			swss.ClearINI_Sections()	
ImgExit_Click(), lblExit_Click()	Form_QueryUnload()			
	Form_Unload()	swss.CleanUpObjects()		
		swss.ClearINI_Sections()		

MODULE: FRMINIT.FRM				
FIRST CALL DEPTH	**SECOND CALL DEPTH**	**THIRD CALL DEPTH**	**FOURTH CALL DEPTH**	**FIFTH CALL DEPTH**
imgFundingDetermination_Click(), lblFundingDetermination_Click()	mnuScreensFundingDetermination_Click()	millerm7.TicklerShells()		
		Form_QueryUnload()		
		Form_Unload()	swss.CleanUpObjects()	
			swss.ClearINI_Sections()	
ImgLegal_Click(), lblLegal_Click()	mnuScreensLegal_Click()	millerm7.TicklerShells()		
		Form_QueryUnload()		
		Form_Unload()	swss.CleanUpObjects()	
			swss.ClearINI_Sections()	
imgMARE_Registration_Click(), lblMARE_Registration_Click()	mnuScreensMARE_Registration_Click()	millerm7.TicklerShells()		
		Form_QueryUnload()		
		Form_Unload()	swss.CleanUpObjects()	
			swss.ClearINI_Sections()	
imgMedicaid_Click(), lblMedicaid_Click()	mnuScreensMedicaid_Click()	millerm7.TicklerShells()		
		Form_QueryUnload()		
		Form_Unload()	swss.CleanUpObjects()	
			swss.ClearINI_Sections()	
imgMedical_Click(), lblMedical_Click()	mnuScreensMedical_Click()	millerm7.TicklerShells()		
		Form_QueryUnload()		
		Form_Unload()	swss.CleanUpObjects()	
			swss.ClearINI_Sections()	
imgMemberInformation_Click(), lblMemberInformation_Click()	mnuScreensMemberInfo_Click()	swss.PrivateINI_Register()		
		swss.tGetAppPath()		
		swss.lPrivateINI_PutString()		
		millerm7.TicklerShells()		
		Form_QueryUnload()		

MODULE: FRMINIT.FRM				
FIRST CALL DEPTH	**SECOND CALL DEPTH**	**THIRD CALL DEPTH**	**FOURTH CALL DEPTH**	**FIFTH CALL DEPTH**
		Form_Unload()	swss.CleanUpObjects()	
			swss.ClearINI_Sections()	
imgPayment_Click()	lblPayment_Click()	fLogID_Change()	millerm7.GetCaseInformation()	
		mnuScreensPaymentBC_Click()	swss.PrivateINI_Register()	
			swss.tGetAppPath()	
			swss.lPrivateINI_PutString()	
			millerm7.TicklerShells()	
			Form_QueryUnload()	
			Form_Unload()	swss.CleanUpObjects()
				swss.ClearINI_Sections()
imgPlacement_Click(), lblPlacement_Click()	mnuScreensPlacement_Click()	millerm7.TicklerShells()		
		Form_QueryUnload()		
		Form_Unload()	swss.CleanUpObjects()	
			swss.ClearINI_Sections()	
imgReportGeneration_Click(), lblReportGeneration_Click()	mnuScreensReportGeneration_Click()	millerm7.TicklerShells()		
		Form_QueryUnload()		
		Form_Unload()	swss.CleanUpObjects()	
			swss.ClearINI_Sections()	
imgUtilities_Click(), lblUtilities_Click()	mnuScreensUtilities_Click()	millerm7.TicklerShells()		
		Form_QueryUnload()		
		Form_Unload()	swss.CleanUpObjects()	
			swss.ClearINI_Sections()	
lblPayment_Click()	fLogID_Change()	millerm7.GetCaseInformation()		
	mnuScreensPaymentBC_Click()	swss.PrivateINI_Register()		
		swss.tGetAppPath()		

MODULE: FRMINIT.FRM				
FIRST CALL DEPTH	**SECOND CALL DEPTH**	**THIRD CALL DEPTH**	**FOURTH CALL DEPTH**	**FIFTH CALL DEPTH**
		swss.lPrivateINI_PutString()		
		millerm7.TicklerShells()		
		Form_QueryUnload()		
		Form_Unload()	swss.CleanUpObjects()	
			swss.ClearINI_Sections()	
mnuScreens5DayPacket_Click(), mnuScreensCaseClosing_Click(), mnuScreensCaseListing_Click(), mnuScreensCaseRegistration_Click(),mnuScreensCaseSummary_Click(), mnuScreensComments_Click(), mnuScreensEducation_Click(), mnuScreensFundingDetermination_Click(), mnuScreensLegal_Click(), mnuScreensMARE_Registration_Click(), mnuScreensMedicaid_Click(), mnuScreensMedical_Click(), mnuScreensProvider_Click(), mnuScreensReportGeneration_Click(),mnuScreensUtilities_Click(), mnuScreensTicklers_Click()	millerm7.TicklerShells()			
	Form_QueryUnload()			
	Form_Unload()	swss.CleanUpObjects()		
		swss.ClearINI_Sections()		
mnuScreensChildInfo_Click(), mnuScreensMemberInfo_Click(), mnuScreensPaymentBC_Click()	swss.PrivateINI_Register()			
	swss.tGetAppPath()			
	swss.lPrivateINI_PutString()			
	millerm7.TicklerShells()			
	Form_QueryUnload()			
	Form_Unload()	swss.CleanUpObjects()		
		swss.ClearINI_Sections()		
TxtLogNo_LostFocus()	fLogID_Change()	millerm7.GetCaseI		

MODULE: FRMINIT.FRM				
FIRST CALL DEPTH	**SECOND CALL DEPTH**	**THIRD CALL DEPTH**	**FOURTH CALL DEPTH**	**FIFTH CALL DEPTH**
		nformation()		

MODULE: millerm7.bas					
FIRST CALL DEPTH	**SECOND CALL DEPTH**	**THIRD CALL DEPTH**	**FOURTH CALL DEPTH**	**FIFTH CALL DEPTH**	**SIXTH CALL DEPTH**
CaseProgramAndStatus()	CheckLoginUserWorkerID()	fSetUpdateMode()	swss.PrivateINI_Register		
			swss.tGetAppPath()		
			swss.lPrivateINI_PutString()		
		fCheckProgramCodeList()	fSetUpdateMode()	swss.PrivateINI_Register	
				swss.tGetAppPath()	
				swss.lPrivateINI_PutString()	
	fCheckSecurityLevel()	CheckLoginUserWorkerID()	fSetUpdateMode()	swss.PrivateINI_Register	
				swss.tGetAppPath()	
				swss.lPrivateINI_PutString()	
			fCheckProgramCodeList()	fSetUpdateMode()	swss.PrivateINI_Register
					swss.tGetAppPath()
					swss.lPrivateINI_PutString()
		fCheckProgramCodeList()	fSetUpdateMode()	swss.PrivateINI_Register	
				swss.tGetAppPath()	
				swss.lPrivateINI_PutString()	

GetCaseInformation()	MainMenu_GetCaseInfo()	swss.makeconnection()			
		{SWSS.Main_Menu.get_case_info}			
		swss.IncrementMessageDots()			
		CallSQL_Procedure()			
	swss.PrivateINI_Register				
	swss.tGetAppPath()				
	swss.lPrivateINI_PutString()				
	swss.IncrementMessageDots()				
	Store_SQL_Time()	swss.PrivateINI_Register			
		swss.tGetAppPath()			
		swss.lPrivateINI_PutString()			
	swss.bPrivateINI_SectionExists()				
	swss.PrivateINI_DeleteSection()				
	swss.ExtractINI_Info()				
	{SWSS.Main_Menu.get_responsibility}				
	swss.FillQueriedResultset()				
	{SWSS.Main_Menu.get_update_worker_ids}				
	GetSecondaryWorkers()	{SWSS.Main_Menu.get_scndry_wrkr_Ids}			
		swss.IncrementMessageDots()			
		swss.FillQueriedResultset()			
		swss.PrivateINI_Register()			
		swss.tGetAppPath()			
		swss.PrivateINI_DeleteSection()			
		swss.lPrivateINI_PutString()			
		Store_SQL_Time()	swss.PrivateINI_Register		

			swss.tGetAppPath()		
			swss.lPrivateINI_PutString()		
		{SWSS.Main_Menu.get_sup_sec_wrkr}			
		{SWSS.Main_Menu.get_co_adpt_supvsr_id}			
	CaseProgramAndStatus()	(see the CaseProgramAndStatus() above in the **First Call Depth** column for its flow structure)			
TicklerShells()	GetCaseInformation()	(see GetCaseInformation()above in the **First Call Depth** column for its flow structure)			
	swss.CommonShell()				

Report (output) Images

Print out versions of each output report generated by the module. For each image, explain its usage.

Not applicable.

Data Elements

For every element on a screen output report:

- Map each data element displayed, printed, or entered to the database table and field, such as the example done by SDM that Paula mentioned in the team meeting (p:\users\share\servwork\SWSS\templates\ReqTemplates\DataDefinitions).xls

Form Field Description	Data Source	Data Target	Target Type	Constraints	Reqd
Main Menu Screen:					
Log #	swss.swss_case.log_id				Y
Case #	swss.swss_case.cis_case_no				
Case Name (tooltip)	swss.swss_case.case_name				
Data Elements not displayed:					
gtResponsibility	swss.load_case.responsibility				Y
gtUserSecurityLevel	swss.security.security_code				Y
gtProgramCode	swss.swss_case.program_code				Y
gtCaseStatus	swss.swss_case.program_status				Y
gtCaseState	swss.swss_case.swss_case_state				Y
gtLegalStatus	swss.swss_case.latest_legal_status				
gtChild_ID	swss.swss_case.child_id				Y

gtLivingArrangement	swss.swss_case.latest_living_arrangement	
gtCaseCounty	swss.swss_case.county_no	
gtWorkerLoadNumber	swss.load_case.load_no	Y
ftPrimaryWorkerID	swss.swss_case.primary_worker_id	Y
gtGoalCode	swss.cfc_info.goal_code	
gtUpdateWorkers (array)	swss.load.worker_id	Y
gtSecondaryWorkers (array)	swss.load_worker.worker_id	Y

- List and discuss any specific validation routines, constraints, or dependent data validations (like legal status and living arrangement) that are not in the data dictionary. You can check (and copy from) the User Requirements Data Element Description for these type of validations.

- Explain the instancing of this data element in laymen's terms. This is implied in the table name, usually, such as the "Case_Person" table refers to an instance of a (group) person record in a particular case, and the "Group_Person" table refers to an instance of a person in a particular sibling group. Go ahead and say it like that, as it applies. This includes "historical" data, such as Medicaid_History, which is an instance of medicaid data over time.

Integration with Existing System

1. The Main Menu module is usually the starting point for all other SWSS modules. The icon layout on its only screen is reflected in the submenu list of the Sections menu in the Common Menu Bar. Selecting icons on this screen takes users to other SWSS modules.
2. This module is also the point at which case information is gathered from the database after a case has been user-selected. The very basic elements of the case needed by most other SWSS modules is then stored into the SWSS_INI.INI file. When each module starts up, this case information is then read and dealt with appropriately.
3. The Main Menu module also acts as the "clean" exit point for users to terminate their SWSS session. If fatal errors don't crash their session, users should be brought back to the Main Menu. They can then select the Exit icon and confirm that they wish to end the current SWSS session.
4. This module also performs some very basic security checking to determine if the current user is to be permitted access to the module he/she requests. If access is permitted, this module assigns a generic "level" of access to the module. This access level is stored in the SWSS_INI.INI file and is retrieved by each SWSS module as it is loaded up.

Module Dependencies

If this module depends on other modules, or if other modules depend on this module, to an extent beyond what has been described in the "Integration with Existing System" section above, such as Medicaid requiring both Placement and Funding data, list and explain those dependencies here.

See "Integration with Existing System" above.

Database Subject Area

```
/*****************************************************************************
*       Gather most data for the SWSS_INI.ini file from the swss_case table.  *
*       Added latest living arrangement output value.                         *
*       Added t_goal_code table output and o_goal_code output parameter.      *
*       Need to bring back high_profile_ind; it will be used to determine if  *
*       the user has access to the case.                                      *
*       Modified the code so that it reads the last name and first name of the *
```

```
*       child and creates the short name (per CM-4.5.1.2.1).                    *
*****************************************************************************/
TYPE SWSS_INI_Data_Rec IS RECORD
(
t_cis_case_number   swss_case.cis_case_no%TYPE,
t_case_name         swss_case.case_name%TYPE,
t_case_program      swss_case.program_code%TYPE,
t_case_status       VARCHAR2(10),
t_legal_status      swss_case.latest_legal_status%TYPE,
t_case_county       swss_case.county_no%TYPE,
t_child_id          swss_case.child_id%TYPE,
t_primary_id        swss_case.primary_worker_id%TYPE,
t_living_arrange    swss_case.latest_living_arrangement%TYPE,
t_primary_load      load_case.load_no%TYPE,
t_goal_code         cfc_info.goal_code%TYPE,
t_high_profile_ind  swss_case.high_profile_ind%TYPE
);

TYPE SWSS_INI_Data_Cursor IS REF CURSOR RETURN SWSS_INI_Data_Rec;

PROCEDURE get_case_info
        (i_log_id             IN    swss_case.log_id%TYPE,
         SWSSINIDataCursor  IN OUT MAIN_MENU.SWSS_INI_Data_Cursor);

/*****************************************************************************************
*       RETRIEVE CASE INFORMATION (NOT FOUND IN THE GET_CASE_INFO                        *
*       PROCEDURE OF THE MISCPKG PACKAGE) BASED UPON A CASE'S LOG                        *
*       NUMBER. PRIMARY WORKER'S LOAD NUMBER PASSED TO PROCEDURE                         *
*       TABLE OF UPDATE CAPABLE WORKER Ids.                                              *
*       THIS PROCEDURE ONLY WORKS IF EACH WORKER IS ONLY ASSIGNED TO                     *
*       ONE SUPERVISOR.  IF THAT IS NOT THE CASE, THIS PROCEDURE MAKE                    *
*       NOT PRODUCE THE CORRECT RESULTS.                                                 *
*****************************************************************************************/
TYPE t_update_worker_ids    IS TABLE OF load_worker.worker_id%TYPE;

TYPE SWSS_INI_Update_Worker_Ids_Rec IS RECORD
(
t_update_worker_ids   load_worker.worker_id%TYPE
);

TYPE SWSS_INI_Update_Worker_Ids_Cur IS REF CURSOR RETURN SWSS_INI_Update_Worker_Ids_Rec;

PROCEDURE get_update_worker_ids
        (i_primary_load_number IN     load_case.load_no%TYPE,
         UpdateWorkerIdsCursor IN OUT MAIN_MENU.SWSS_INI_Update_Worker_Ids_Cur );

/*****************************************************************************
*       GET LIST OF SECONDARY WORKER Ids                                     *
*****************************************************************************/
TYPE SWSS_INI_Secondary_Id_Rec IS RECORD
(
t_secondary_id  load_worker.worker_id%TYPE
);

TYPE SWSS_INI_Secondary_Id_Cursor IS REF CURSOR RETURN SWSS_INI_Secondary_Id_Rec;
```

```
PROCEDURE get_scndry_wrkr_IDs
        (i_log_id         IN     swss_case.log_id%TYPE,
         SecondaryIdCursor IN OUT MAIN_MENU.SWSS_INI_Secondary_Id_Cursor);

/*********************************************************************************************
*       GET THE WORKER_ID OF THE SECONDARY WORKER'S SUPERVISOR                                *
*       This is the first procedure "broken out" of the get_scndry_wrkr_IDs procedure.        *
*       It returns the worker_id of the supervisor of the secondary worker assigned to        *
*       the case, if that worker has a supervisor.                                            *
*********************************************************************************************/
PROCEDURE get_sup_sec_wrkr
        (i_log_id         IN     swss_case.log_id%TYPE,
         o_sup_wrkr_id    OUT    load_worker.worker_id%TYPE);

/*********************************************************************************************
*       GET THE WORKER_ID OF THE COUNTY ADOPTION SUPERVISOR                                   *
*       This is the second procedure "broken out" of the get_scndry_wrkr_IDs                  *
*       procedure.  It returns the worker_id of the county adoption supervisor                *
*       in the secondary worker's county.                                                     *
*********************************************************************************************/
/*********************************************************************************************/
/*  Added this block 04/18/2000 - MDM                                                        */
TYPE Co_ADPT_Sup_List_Rec IS RECORD
(
t_Co_ADPT_Sup_id  load_worker.worker_id%TYPE
);

TYPE Co_ADPT_Sup_Cursor IS REF CURSOR RETURN Co_ADPT_Sup_List_Rec;
/*********************************************************************************************/
PROCEDURE get_co_adpt_supvsr_id
        (i_log_id         IN     swss_case.log_id%TYPE,
   /*****************************************************/
   /*  Changed this line of code 04/18/2000 - MDM       */
         --o_sup_wrkr_id    OUT    load_worker.worker_id%TYPE);
    CoADPT_SupCursor  IN OUT  MAIN_MENU.Co_ADPT_Sup_Cursor);
   /*****************************************************/

/*****************************************************************************
*       GET LOGIN USER'S RESPONSIBILITY TO SELECTED CASE.                    *
*****************************************************************************/
PROCEDURE get_responsibility
        (i_log_id         IN  load_case.log_id%TYPE,
         i_login_worker_id IN  load_worker.worker_id%TYPE,
         o_responsibility  OUT load_case.responsibility%TYPE,
         o_security_level  OUT security.security_code%TYPE
        );

/*********************************************************************************************
*       Gets all load numbers and their corresponding security codes for a                    *
*       particular worker_id.                                                                 *
*********************************************************************************************/
TYPE SWSS_INI_Security_Level_Rec IS RECORD
(
t_load_number     load_worker.load_no%TYPE,
t_security_level  security.security_code%TYPE
);
```

```
TYPE SWSS_INI_Sec_Level_Cur IS REF CURSOR RETURN SWSS_INI_Security_Level_Rec;

PROCEDURE get_load_securities
        (i_login_worker_id IN  load_worker.worker_id%TYPE,
         SWSSINISecurityLevelCursor IN OUT MAIN_MENU.SWSS_INI_Sec_Level_Cur
        );
```

Data Warehouse

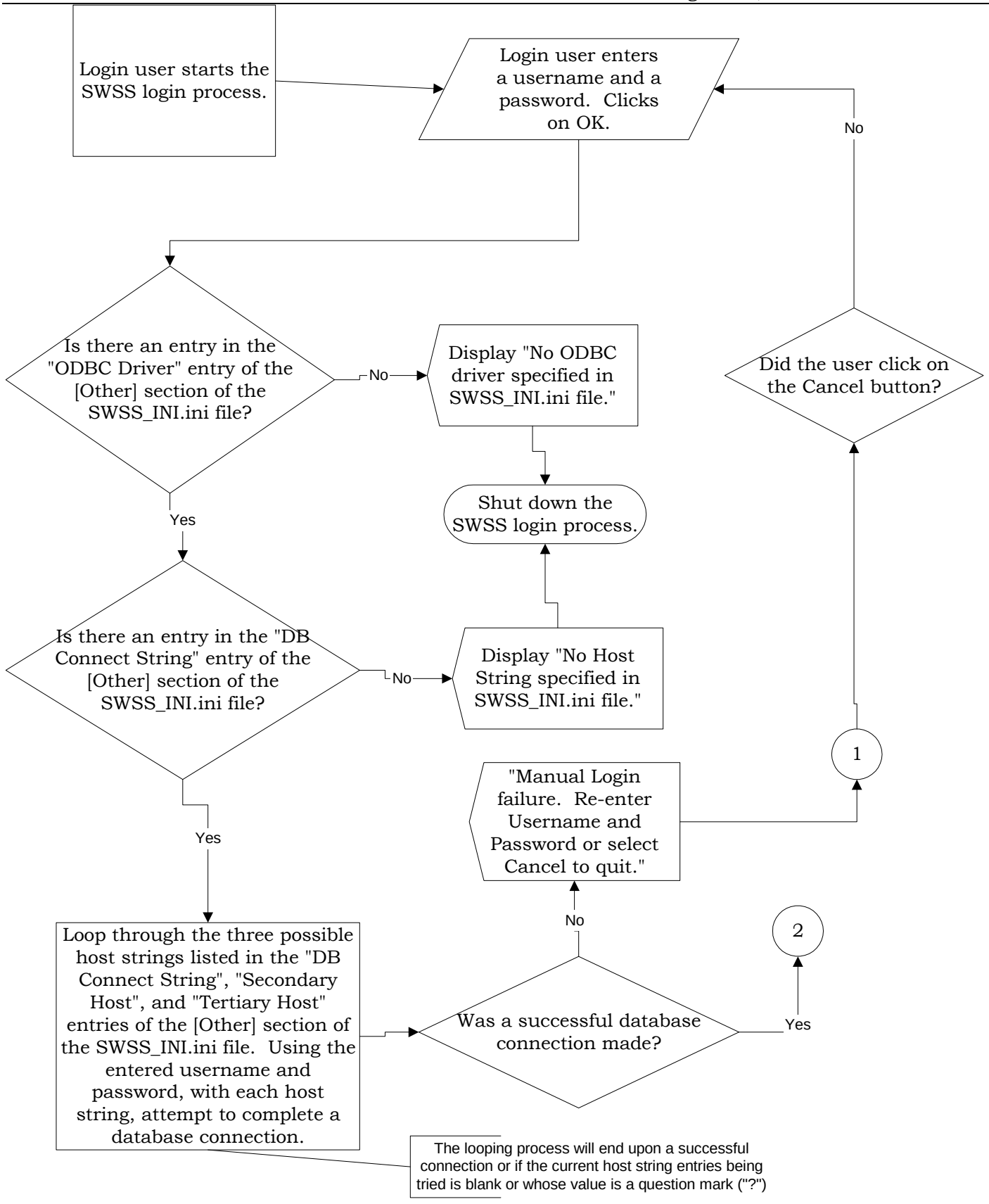
If known, explain which items are added to the data warehouse and under what conditions they are written. Hopefully we can reference a document or set of documents supported by the data warehouse.

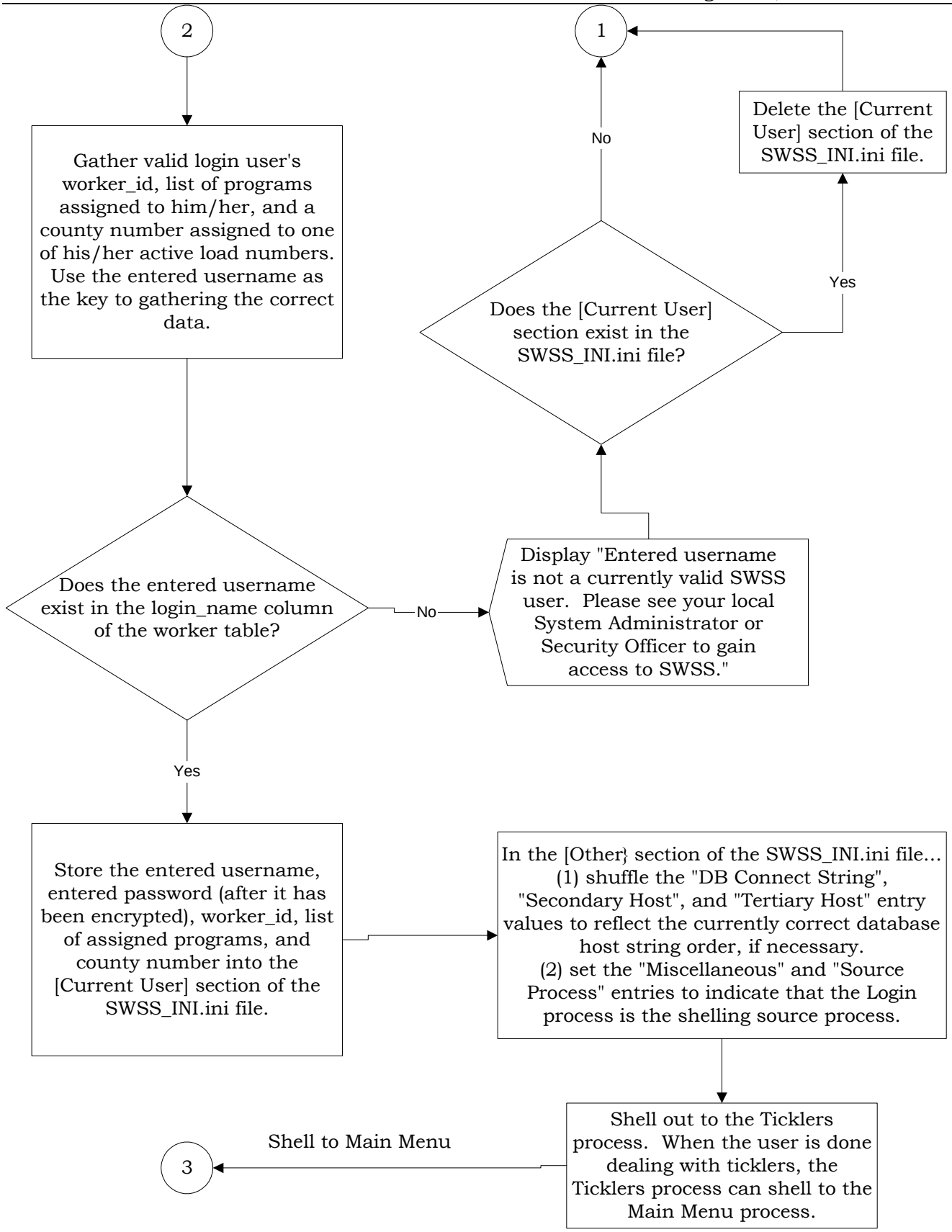
Not Applicable.

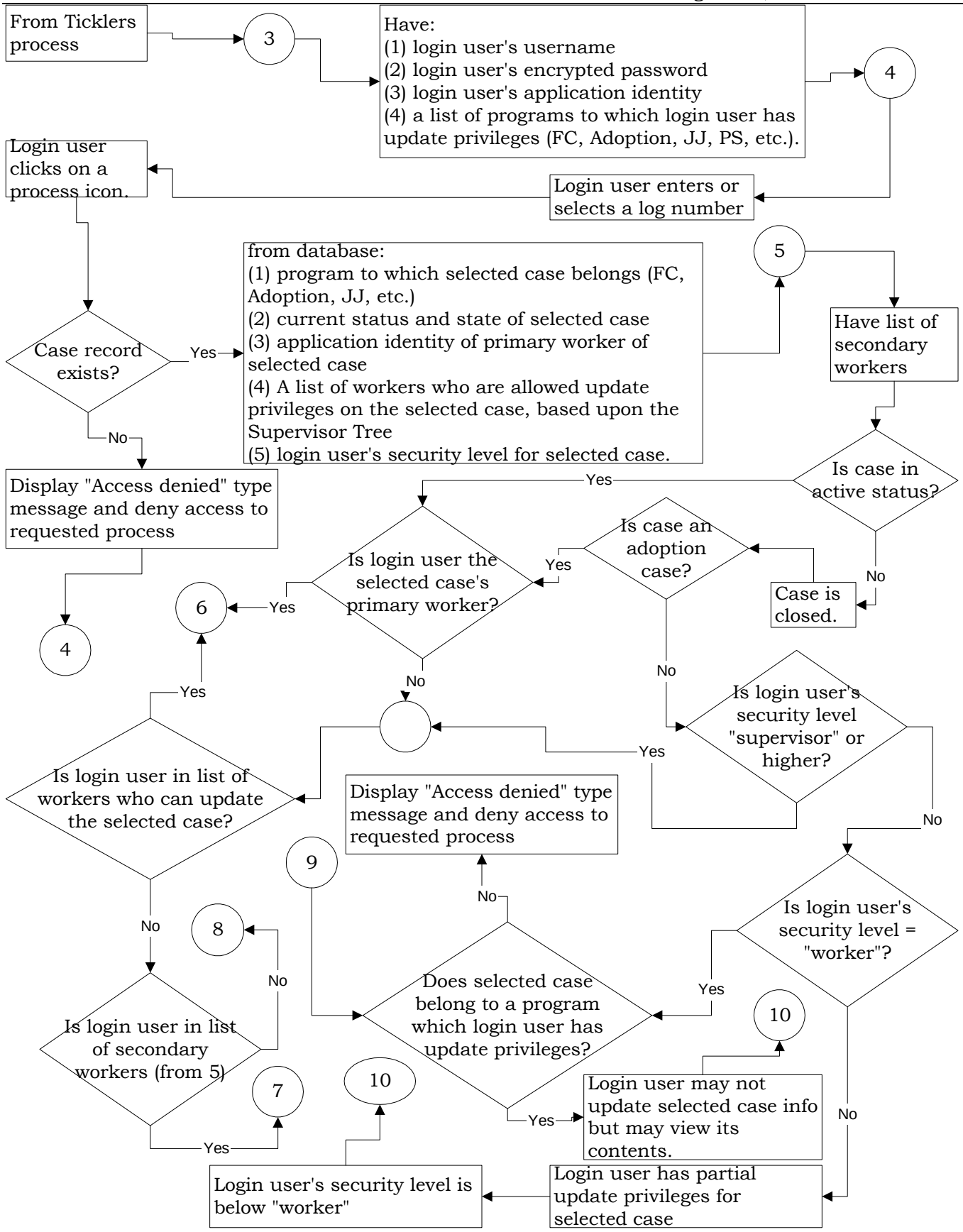
Technical Issues

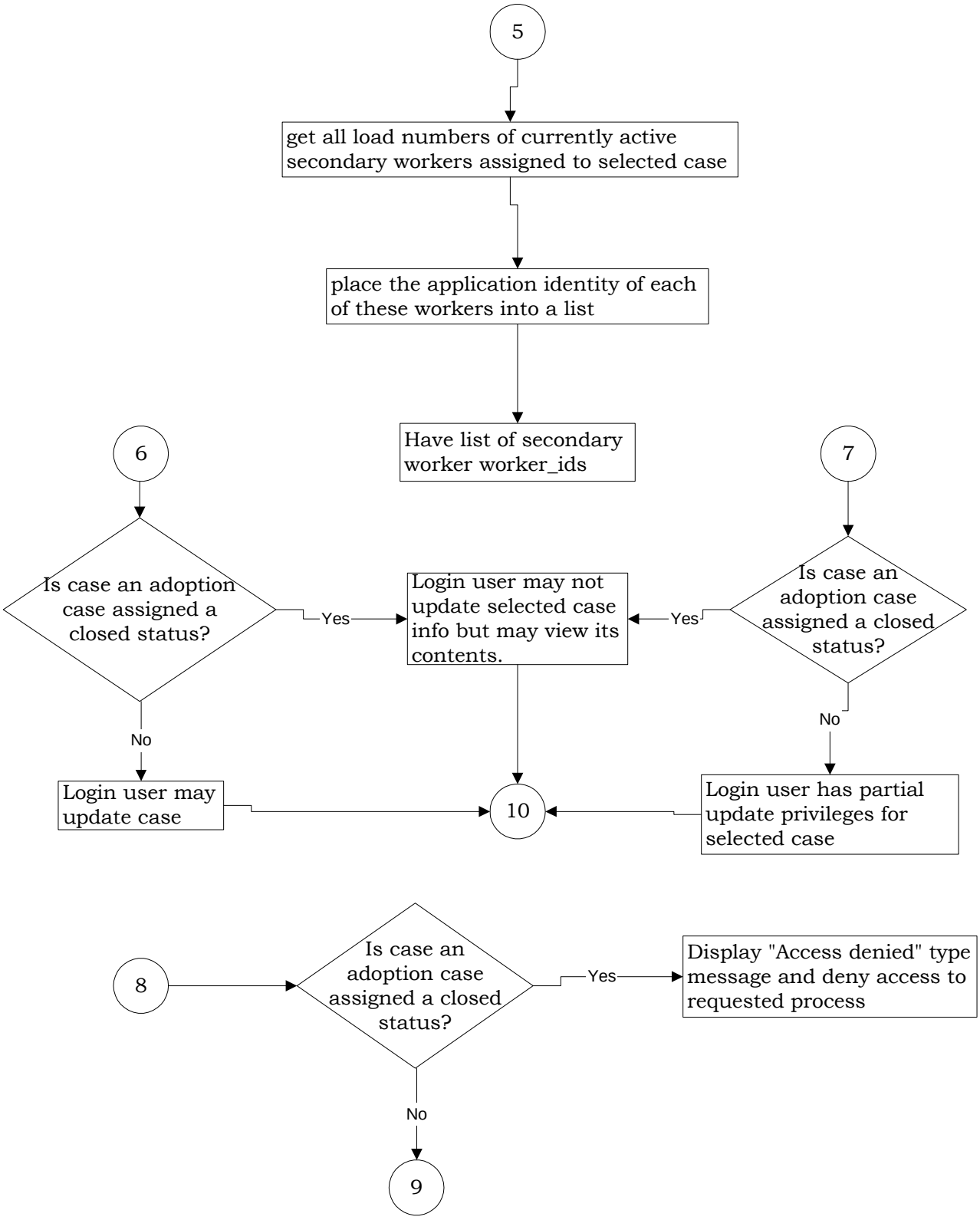
Discuss any tricky things that happen in the module that someone who maintains the application may not recognize at first glance. Sibling group sharing, legal status switches, or reusing person Ids.

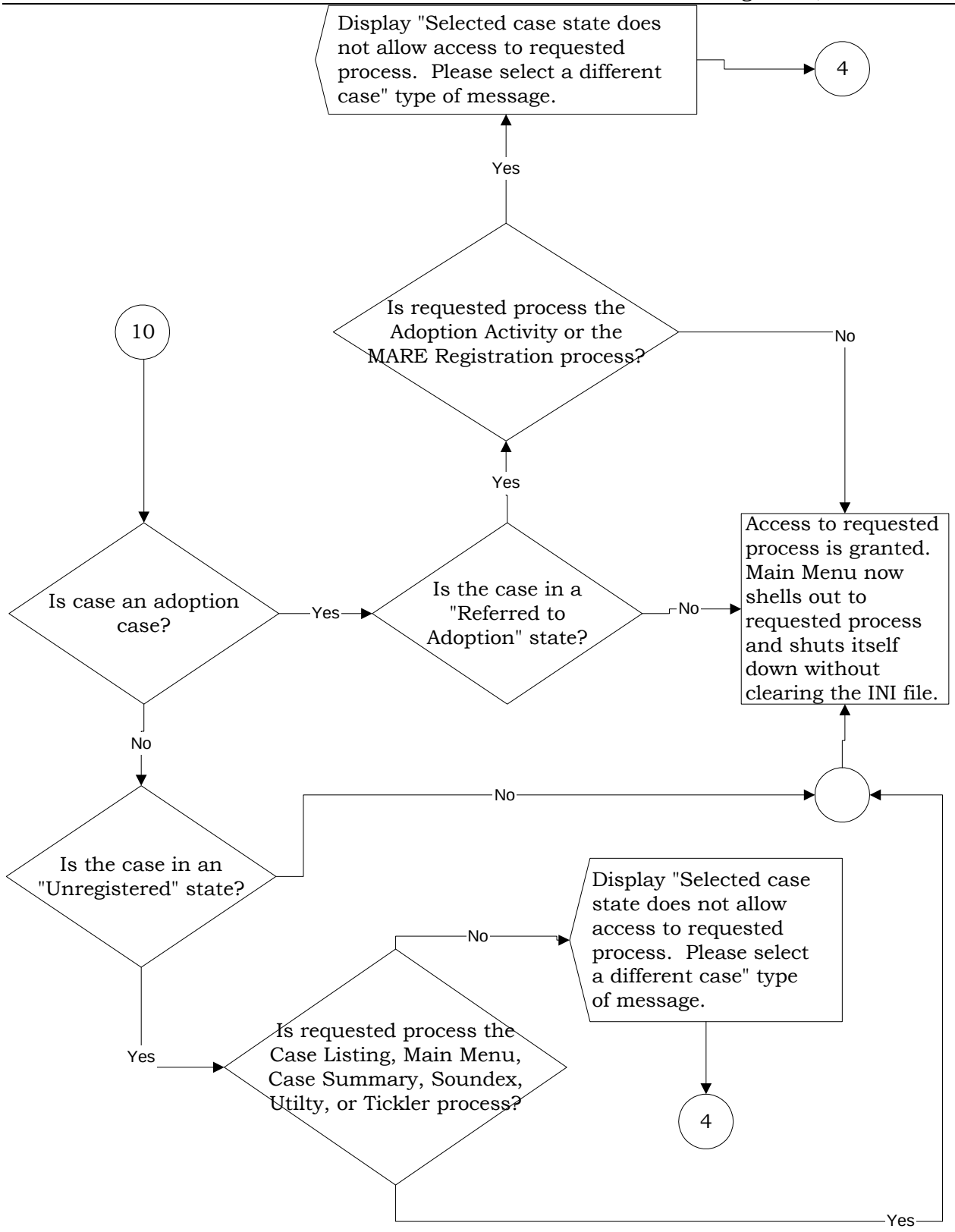
1. The MouseMove event is used by almost every control on this screen to control the “selection borders” for the different module icons. As the user moves the mouse pointer from one icon to another, the MouseMove code turns the “selection borders” on and off for the appropriate icon-label control combinations. This “selection border” code is also used on the “Main” screen in the Report Generation module.
2. The module permission’s code is exceeding complex and is better explained by looking at the flowcharts on the following five pages.


Login user starts the SWSS login process.
Login user enters a username and a password. Clicks on OK.
No
Is there an entry in the "ODBC Driver" entry of the [Other] section of the SWSS_INI.ini file?
No
Display "No ODBC driver specified in SWSS_INI.ini file."
Did the user click on the Cancel button?
Shut down the SWSS login process.
Yes
Is there an entry in the "DB Connect String" entry of the [Other] section of the SWSS_INI.ini file?
No
Display "No Host String specified in SWSS_INI.ini file."
1
"Manual Login failure. Re-enter Username and Password or select Cancel to quit."
Yes
Loop through the three possible host strings listed in the "DB Connect String", "Secondary Host", and "Tertiary Host" entries of the [Other] section of the SWSS_INI.ini file. Using the entered username and password, with each host string, attempt to complete a database connection.
No
2
Was a successful database connection made?
Yes
The looping process will end upon a successful connection or if the current host string entries being tried is blank or whose value is a question mark ("?")


2
1
Gather valid login user's worker_id, list of programs assigned to him/her, and a county number assigned to one of his/her active load numbers. Use the entered username as the key to gathering the correct data.
No
Delete the [Current User] section of the SWSS_INI.ini file.
Yes
Does the [Current User] section exist in the SWSS_INI.ini file?
Does the entered username exist in the login_name column of the worker table?
No
Display "Entered username is not a currently valid SWSS user. Please see your local System Administrator or Security Officer to gain access to SWSS."
Yes
Store the entered username, entered password (after it has been encrypted), worker_id, list of assigned programs, and county number into the [Current User] section of the SWSS_INI.ini file.
In the [Other} section of the SWSS_INI.ini file... (1) shuffle the "DB Connect String", "Secondary Host", and "Tertiary Host" entry values to reflect the currently correct database host string order, if necessary. (2) set the "Miscellaneous" and "Source Process" entries to indicate that the Login process is the shelling source process.
Shell out to the Ticklers process. When the user is done dealing with ticklers, the Ticklers process can shell to the Main Menu process.
Shell to Main Menu
3


From Ticklers process
3
Have:
(1) login user's username
(2) login user's encrypted password
(3) login user's application identity
(4) a list of programs to which login user has update privileges (FC, Adoption, JJ, PS, etc.).
4
Login user clicks on a process icon.
Login user enters or selects a log number
Case record exists?
Yes
No
from database:
(1) program to which selected case belongs (FC, Adoption, JJ, etc.)
(2) current status and state of selected case
(3) application identity of primary worker of selected case
(4) A list of workers who are allowed update privileges on the selected case, based upon the Supervisor Tree
(5) login user's security level for selected case.
5
Have list of secondary workers
Display "Access denied" type message and deny access to requested process
4
Is case in active status?
Yes
No
Is case an adoption case?
Yes
No
Case is closed.
Is login user the selected case's primary worker?
Yes
6
Yes
No
Is login user's security level "supervisor" or higher?
Yes
No
Is login user in list of workers who can update the selected case?
Yes
No
Display "Access denied" type message and deny access to requested process
9
8
No
No
Is login user's security level = "worker"?
Yes
No
Is login user in list of secondary workers (from 5)
No
Yes
7
Does selected case belong to a program which login user has update privileges?
Yes
No
10
10
Login user may not update selected case info but may view its contents.
Login user's security level is below "worker"
Login user has partial update privileges for selected case


5
get all load numbers of currently active secondary workers assigned to selected case
place the application identity of each of these workers into a list
Have list of secondary worker worker_ids
6
Is case an adoption case assigned a closed status?
Yes
Login user may not update selected case info but may view its contents.
Yes
7
Is case an adoption case assigned a closed status?
No
No
Login user may update case
10
Login user has partial update privileges for selected case
8
Is case an adoption case assigned a closed status?
Yes
Display "Access denied" type message and deny access to requested process
No
9


Display "Selected case state does not allow access to requested process. Please select a different case" type of message.
4
Yes
Is requested process the Adoption Activity or the MARE Registration process?
No
10
Yes
Is case an adoption case?
Yes
Is the case in a "Referred to Adoption" state?
No
Access to requested process is granted. Main Menu now shells out to requested process and shuts itself down without clearing the INI file.
No
No
Is the case in an "Unregistered" state?
Display "Selected case state does not allow access to requested process. Please select a different case" type of message.
No
Yes
Is requested process the Case Listing, Main Menu, Case Summary, Soundex, Utilty, or Tickler process?
4
Yes

Test Plans

Include the test plan developed for this module, and references to any scenarios that apply to it.

Test Plan – SWSS Main Menu

Matthew D. Miller

Process Accessibility

- All workers are allowed access to the Main Menu process.

Case Functionality

- Most process selections from the Main Menu require that a case be selected before attempting to access the process. If the worker attempts to access one of these process without selecting a case, a message should appear allowing the choice of either executing Case Listing to select a case or canceling the process activation request. The only processes which do not require a case are:
  - Case Listing
  - Find a Client
  - Add New
  - Ticklers
  - Utilities (special note: The Reconciliation process is accessible through Utilities and ***does*** require a case. Therefore, if the worker wishes to successfully get to Reconciliation through Utilities, they need to specify a case from Main Menu or Case Listing first)
- A case can be selected by either entering a log number in the "Log #" field, by selecting a case from the Case Listing process, or by selecting a case from the Find a Client (or Soundex) process. When the Case Listing or Find a Client process returns to the Main Menu, the log number selected in that process should appear in the Log # field of the Main Menu. Selecting a case by specifying only a Case number is currently not supported (this functionality should be accomplished in the Find a Client process).
- The Main Menu process should only have to gather case information from the Oracle database if:
  - Main Menu is activated from the Case Listing, Find A Client, or Case Registration processes
  - The log number has been changed in the Log # field (except if that field has been cleared) and the worker has not requested the activation of the Case Listing, Find a Client, or Add New processes
- If a log number is entered which does not exist in the system, a message should be displayed saying so.
- Two processes on the Main Menu require that the selected case be an Adoption case before they may be activated. These processes are Adoption Activity and M.A.R.E. Registration. If the selected case is not an adoption case and one of these two processes

is selected, a message should pop up informing the worker that only adoption cases may be modified through this process.

- The Funding Determination process requires that the selected case is a Foster Care case or an assigned Adoption case without an adoption placement before it can be activated. If the selected case does not fit these criteria, a message box should pop up explaining what the problem is.
- A case's program type should be verifiable by activating the Case Summary process.
- After the worker selects a case and selects a process, the case information should then be stored into the SWSS_INI.INI file. This information includes the worker's security level with respect to the case, the worker's responsibility to the case (if any), and a list of workers who are allowed to update the selected case.
- A worker's "SWSS session" should be terminated by exiting from the Main Menu process by selecting either the Exit icon or by selecting the Exit submenu in the File menu of the menu bar. Upon this session termination, all section of the SWSS_INI.INI file should be cleared except for the [Other] section.

Case Update Determination

- If the case is an Adoption case
  - And if the worker is the primary worker assigned to this case or is in the list of update capable worker for the selected case (the update capable list consists of the alternate worker assigned to the primary worker of the case, the supervisor assigned to the primary worker of the case, the alternate worker assigned to the supervisor of the primary worker of the case, that supervisor's supervisor, etc)
    - And if the case is closed
      - The worker should have view-only privileges on the case information.
    - And if the case not closed but is active
      - The worker should have full update privileges on the case information.
  - And if the worker is not the primary worker assigned to this case and is not in the list of update capable worker for the selected case, but is a secondary worker assigned to the selected case
    - And if the case is closed
      - The worker should have view-only privileges on the case information.
    - And if the case is not closed but is active
      - The worker should have restricted update privileges on the case information. Whichever process the worker activates will determine what those update restrictions are for the selected case information.
  - And if the worker is not the primary worker, an update capable worker, or a secondary worker to the case
    - And if the case is closed
      - The worker should be denied access to the case information. This should be told to the worker with a message box.
    - And if the case is not closed but is active
      - And if the worker has been assigned to the Adoption program
        - The worker should have view-only privileges on the case information.

- And if the worker has not been assigned to the Adoption program
  - The worker should be denied access to the case information. This should be told to the worker with a message box.

- Else if the case is not an Adoption case but is a Foster Care, Protective Services, Juvenile Justice, or Referral case
  - And if the case is active
    - And if the worker is the primary worker assigned to this case or is in the list of update capable worker for the selected case (the update capable list consists of the alternate worker assigned to the primary worker of the case, the supervisor assigned to the primary worker of the case, the alternate worker assigned to the supervisor of the primary worker of the case, that supervisor's supervisor, etc)
      - The worker should have full update privileges on the case information.
    - And if the worker is not the primary worker assigned to this case and is not in the list of update capable worker for the selected case, but is a secondary worker assigned to the selected case
      - The worker should have restricted update privileges on the case information. Whichever process the worker activates will determine what those update restrictions are for the selected case information.
    - And if the worker is not the primary worker, an update capable worker, or a secondary worker to the case
      - And if the worker has been assigned to the same program that the case belongs to
        - The worker should have view-only privileges on the case information.
      - And if the worker has not been assigned to the program that the case belongs to
        - The worker should be denied access to the case information. This should be told to the worker with a message box.
  - And if the case is not active but is closed
    - And if the worker has been assigned the supervisor security level or higher (00 – 30)
      - And if the worker is in the list of update capable workers for the selected case (the update capable list consists of the alternate worker assigned to the primary worker of the case, the supervisor assigned to the primary worker of the case, the alternate worker assigned to the supervisor of the primary worker of the case, that supervisor's supervisor, etc)
        - The worker should have full update privileges on the case information.
      - And if the worker is not the primary worker assigned to this case and is not in the list of update capable worker for the selected case, but is a secondary worker assigned to the selected case
        - The worker should have restricted update privileges on the case information. Whichever process the worker activates will determine what those update restrictions are for the selected case information.
      - And if the worker is not an update capable worker or a secondary worker for the selected case

        - The worker should be denied access to the case information. This should be told to the worker with a message box.
    - And if the worker has been assigned the worker security level (31 – 60)
        - And if the worker has been assigned to the same program that the case belongs to
            - The worker should have view-only privileges on the case information.
        - And if the worker has not been assigned to the program that the case belongs to
            - The worker should be denied access to the case information. This should be told to the worker with a message box.
    - And if the worker has been assigned to a security level below the worker level (61 - 99)
        - The worker should have restricted update privileges on the case information. Whichever process the worker activates will determine what those update restrictions are for the selected case information.

Corrections Mode

- If the worker is only allowed view-only privileges on the selected case information
    - The Corrections menu should be disabled and the user should not be allowed to make changes to corrections fields for the case. A message box should also pop up to inform the worker of this.
- Else if the worker is allowed restricted or full update privileges on the case information
    - And if the case is not closed
        - The worker should be able to enter Corrections information for the selected case.
    - And if the case is closed and the worker has been assigned a Supervisor or better security level (00 – 30)
        - The worker should be able to enter Corrections information for the selected case.
    - And if the case is closed but the worker's security level is less than Supervisor level (31 – 99)
        - The Corrections menu should be disabled and the user should not be allowed to make changes to corrections fields for the case. A message box should also pop up to inform the worker of this.

Case Status and Case State

The Main Menu

Case Program and Status outline

1) The case is a CFC, CPS, JJ, or Referral case
    a) The Case Status is Active
        i) Go to **Check Supervisor Tree** outline.
    b) The Case Status is Closed
        i) Go to **Check Security Level** outline.
    c) The Case Status is neither Active or Closed.
        i) Display a message stating that the Status code assigned to the selected case is undefined.
2) The case is an ADPT case

a) The Case Status is either Active or Closed
   i) Go to **Check Supervisor Tree** outline.
b) The Case Status is neither Active or Closed
   i) Display a message stating that the Status code assigned to the selected case is undefined.

3) The case does not belong to either CPS, CFC, ADPT, JJ, or Referral
   a) Display a message stating that security issues have not yet been defined for the selected Case Program.

Check Supervisor Tree outline.

1) The current worker is either the Primary Worker assigned to the case or is a worker who has update permissions on the selected case according to the rules of the Supervisor Tree. This may mean that the current worker is the Primary Worker, the Alternate Worker assigned to the Primary Worker, the Primary Worker's supervisor, the Alternate Worker assigned to the Primary Worker's supervisor, the supervisor of the Primary Worker's supervisor, the Alternate worker assigned to the supervisor of the Primary Worker's supervisor, etc.
   a) The selected case is an Adoption case and the Case Status is Closed.
      i) Set the SWSS_INI.INI Update Mode entry to *Not Updateable*.
   b) The selected case is either not an Adoption case and/or the Case Status is not Closed.
      i) Set the SWSS_INI.INI Update Mode entry to *Updateable*.
2) The current worker is a Secondary Worker assigned to the selected case.
   a) The selected case is an Adoption case and the Case Status is Closed.
      i) Set the SWSS_INI.INI Update Mode entry to *Not Updateable*.
   b) The selected case is either not an Adoption case and/or the Case Status is not Closed.
      i) Set the SWSS_INI.INI Update Mode entry to *Selectively Updateable*. This means that the shelled executable may have to decide how much access the current worker has to case information.
3) The current worker does not fall within the Supervisor Tree for determining access to the selected case, nor is the current worker the Secondary Worker assigned to the case.
   a) The selected case is an Adoption case and the Case Status is Closed.
      i) Display message stating that access to the selected executable is denied and explain why.
   b) The selected case is either not an Adoption case and/or the Case Status is not Closed.
      i) Go to **Check Program Code List** outline.

Check Security Level outline

1) The current worker's security level is Supervisor or better (a security code value ranging from 00 – 30)
   a) Go to **Check Supervisor Tree** outline.
2) The current worker's security level is Worker (a security code value ranging from 31 – 60).
   a) Go to **Check Program Code List** outline
3) The current worker's security level is below Worker level (a security code value ranging from 61 – 99)
   a) Set the SWSS_INI.INI Update Mode entry to *Selectively Updateable*. This means that the shelled executable may have to decide how much access the current worker has to case information.

Check Program Code List outline

1) The selected case belongs to a program which the current worker has been assigned to.
   a) Set the SWSS_INI.INI Update Mode entry to *Not Updateable*
2) The selected case belongs to a program which the current has *not* been assigned to.
   a) Deny access to the selected process and displaay a message stating so and why.

Shelling from one application to another

Corrections Mode outline

1) The Update Mode entry in the SWSS_INI.INI file does not equal *Not Updateable* and a security level has been determined for the current worker on the selected case.
   a) The Case Status is Closed and the current worker is a Supervisor (security code value is <= 30).
      i) The current worker may use the Corrections menu.

b) The Case Status is Closed and the current worker is not a Supervisor (security code value is > 30).
   i) Corrections Mode is deactivated and the Corrections menu is disabled.
c) The Case Status is not Closed.
   i) The Case State is Registered.
      (1) Corrections Mode is deactivated and the Corrections menu is disabled.

2) The Update Mode entry in the SWSS_INI.INI file is set to *Not Updateable* or a security level for the current worker on the current case has not been selected.
   a) Corrections Mode is deactivated and the Corrections menu is disabled.

Determine Permission To Shell outline

1) The current worker is shelling to the Report Generation, Data Reply, 5 Day Packet, Case Closing, Case Listing, Medicaid, Medical, Payment, Placement, Print 133, Print 5S, Providers, Education, Case Summary, Ticklers, Utilities, Main Menu, or the Victim Letter process.
   a) Access to the selected process is granted up to this point.
   b) Go to **Special Shelling Determination** outline.
2) The current worker is shelling to the Adoption Activity or the M.A.R.E. Registration process.
   a) The selected case is an Adoption case and the current worker has permission to use the selected process.
      i) Access to the selected process is granted up to this point.
      ii) Go to **Special Shelling Determination** outline.
   b) The selected case is not an Adoption case.
      i) Access to the selected process is denied. A message stating this is displayed.
3) The current worker is shelling to the Case Registration process.
   a) A log number has been entered into the log number field and the currently running process is the Main Menu process.
      i) The value of the Miscellaneous entry in the SWSS_INI.INI file is set to "Edit"
   b) Access is to the Case Registration process granted up to this point.
   c) Go to **Special Shelling Determination** outline.
4) The current worker is shelling to the Comments process.
   a) The worker has activated the Comments process from the Common Menu Bar.
      i) The value of the Miscellaneous entry in the SWSS_INI.INI file is cleared.
   b) Go to **Special Shelling Determination** outline.
5) The current worker is shelling to the Funding Determination process.
   a) The selected case is a Foster Care case and the user has permission to shell to this process up to this point.
      i) Access to Funding Determination is granted up to this point.
      ii) Go to **Special Shelling Determination** outline.
   b) The selected case is an Adoption case, the Case State is either "Referred" or "Accepted", and the user has permission to shell to this process up to this point.
      i) Access to Funding Determination is granted up to this point.
      ii) Go to **Special Shelling Determination** outline..
   c) The selected case is a Juvenile Justice case, the latest legal status is either "Court Ward-Delinquent", "St Ward-DEL-ACT 150", or "Dual Wardship" (40, 46, or 52 respectively), and the user has permission to shell to this process up to this point.
      i) Access to Funding Determination is granted up to this point.
      ii) Go to **Special Shelling Determination** outline.
   d) The selected case is an Adoption case, the Case State is "Assigned", and a Latest Living Arrangement has been determined.
      i) The Latest Living Arrangement is "Adoptive Home" (code = 04).
         (1) Access to Funding Determination is denied.
      ii) The Latest Living Arrangement is not "Adoptive Home" (code = 04).
         (1) Access to Funding Determination is granted up to this point.
         (2) Go to **Special Shelling Determination** outline.
   e) The current circumstances do not fall under 5a, 5b, 5c, or 5d of this section of this outline.
      i) Access to Funding Determination is denied.
6) The current worker is shelling to either the Member Information or Child Information processes.

a) If permission for shelling to the selected process has been granted up to this point, then access to the selected process is granted up to this point.
b) Go to **Special Shelling Determination** outline.

7) The current worker is shelling to the Legal process.
   a) The selected case is either a Foster Care, Adoption, or Juvenile Justice case.
      i) Access to Legal is granted up to this point.
      ii) Go to **Special Shelling Determination** outline.
   b) The selected case is not a Foster Care, Adoption, or Juvenile Justice case.
      i) Access to Legal is denied.
8) The current worker is shelling to the Retrieve Five Day Packet process.
   a) The selected case is either a Foster Care, Juvenile Justice, or Adoption case.
      i) The Case State is either "Registered" or "Unregistered"
         (1) Access to the Retrieve Five Day Packet process is granted up to this point.
         (2) Go to **Special Shelling Determination** outline.
      ii) The Case State is not "Registered" or "Unregistered"
         (1) Access to the Retrieve Five Day Packet process is denied.
   b) The selected case is not a Foster Care, Adoption, or Juvenile Justice case.
      i) Access to the Retrieve Five Day Packet process is denied.
9) The current circumstances of the current SWSS session do not fit in any of the cases presented in 1-8 of this outline.
   a) Go to **Special Shelling Determination** outline.

Special Shelling Determination outline.

1) The selected case is an Adoption case.
   a) The Case State is "Referred"
      i) The current worker is shelling to the Adoption Activity or the M.A.R.E. Registration process.
         (1) Access to the selected process is denied.
      ii) The current worker is not shelling to the Adoption Activity or the M.A.R.E. Registration process.
         (1) Access to the selected process is granted.
   b) The Case State is not "Referred"
      i) Access to the selected process is granted.
2) The selected case is not an Adoption Case.
   a) The Case State is "Unregistered"
      i) The current worker is shelling to the Case Listing, Case Summary, Soundex, Utilities, Main Menu, Ticklers, Case Registration, or Print 133 processes.
         (1) Access to the selected process is granted.
      ii) The current worker is shelling to a process not listed in (2ai) of this outline.
         (1) Access to the selected process is denied.
   b) The Case State is not "Unregistered"
      i) Access to the selected process is granted.